

## CLAIMS

What is claimed is:

1. A method comprising:
  - receiving a unique key;
  - searching a hash for a match to the unique key;
  - searching a cache for the match to the unique key concurrently with the searching the hash for the match to the unique key;
  - and
  - obtaining information regarding the unique key.
2. The method of claim 1 wherein the unique key is associated with a subscriber of a data packet.
3. The method of claim 1 wherein the hash comprises buckets, each bucket comprising a plurality of locations.
4. The method of claim 3 wherein a first location in the bucket comprises a most recent key.
5. The method of claim 4 wherein the most recent key is a most recently searched key.

6. The method of claim 4 wherein the most recent key is a last added key.
7. The method of claim 1 wherein the cache comprises least recently used keys.
8. A method comprising:
  - rotating entries stored in a plurality of locations of a hash bucket to empty a first location; and
  - adding a key to the first location of the plurality of locations of the hash bucket.
9. The method of claim 8 wherein the rotating the entries comprises moving an entry from a location to a next location of the plurality of locations.
10. The method of claim 9 further comprising moving an entry from a last location to a cache.
11. The method of claim 10 further comprising assigning a timestamp to the entry from the last location.
12. The method of claim 11 further comprising removing an oldest entry from the cache based on the timestamp.

13. The method of claim 11 further comprising removing an oldest entry from the cache based on a total amount of entries in the cache.
14. The method of claim 13 further comprising comparing the total number of entries in the cache to a predetermined maximum number.
15. The method of claim 8 wherein the rotating entries comprises placing a second to a most recent added entry in a second location and a least recent entry from entries remaining in the hash bucket in a last location.
16. A method comprising:
  - receiving a unique key to be deleted;
  - searching a plurality of locations of a hash bucket for a match to the unique key;
  - deleting the unique key from a location of the plurality of locations in the hash bucket upon finding the match to the unique key in the location; and
  - rotating remaining entries of the hash bucket after the deleting the unique key from the location of the plurality of locations in the hash bucket.

17. The method of claim 16 further comprising searching a cache for the match to the unique key concurrently with the searching the plurality of locations of the hash bucket for the match to the unique key.
18. The method of claim 17 further comprising deleting the unique key from the cache upon finding the match to the unique key in the cache.
19. The method of claim 16 wherein the rotating the remaining entries of the hash bucket comprises shifting entries in locations following a deleted unique key location to place an entry stored in a location next to the deleted unique key location into the deleted unique key location.
20. A method comprising:
  - associating an entry of a plurality of entries in a cache with a timestamp;
  - incrementing the timestamp; and
  - deleting the entry of the plurality of entries in the cache based on a value of the timestamp.
21. The method of claim 20 wherein the deleting the entry of the plurality of entries in the cache is performed upon the timestamp reaching a predetermined maximum value.

22. The method of claim 20 wherein the deleting the entry of the plurality of entries in the cache is performed when the timestamp is greater than timestamps of remaining entries of the plurality of entries and a number of total entries in the cache is equal to a predetermined maximum number of entries.

23. A method comprising:

identifying a subscriber associated with a particular packet  
utilizing a line card; and  
transmitting the particular packet to one of a plurality of data cards  
associated with the identified subscriber.

24. The method of claim 23 further comprising identifying the subscriber associated with the particular packet utilizing at least one of the plurality of data cards if the identifying of the subscriber utilizing the line card fails.

25. The method of claim 23 wherein the identifying the subscriber comprises:

generating a search key from the particular packet unique to the subscriber; and  
concurrently searching a first memory storing most recently used keys and a second memory storing least recently used keys for a matching key.

2025-10-01 10:00:00

26. The method of claim 25 further comprising determining an address of the matching key on one of the plurality of data cards.
27. The method of claim 25 wherein the first memory comprises a plurality of buckets comprising a plurality of locations.
28. The method of claim 27 further comprising deleting the matching key from a location in a bucket of the first memory.
29. The method of claim 28 further comprising shifting entries in locations following a deleted matching key location to place an entry stored in a location next to the deleted matching key location into the deleted matching key location.
30. The method of claim 25 further comprising deleting the matching key from the second memory.
31. The method of claim 27 further comprising adding a new key to the first memory by placing a new key in a first location of the plurality of locations in a bucket.

32. The method of claim 31 further comprising rotating entries in remaining locations of the plurality of locations in the bucket to empty the first location of the plurality of locations.
33. The method of claim 32 further comprising moving an entry from a last location of the plurality of locations in the bucket to the second memory.
34. The method of claim 25 further comprising assigning a timestamp to each least recently key in the second memory.
35. The method of claim 34 further comprising deleting a key from the second memory based on the timestamp.
36. The method of claim 35 wherein the deleting the key from the second memory is performed when the timestamp is equal to a predetermined maximum value.
37. The method of claim 35 wherein the deleting the key from the second memory further comprises deleting the key with an oldest timestamp upon a total number of keys in the second memory reaching a predetermined maximum value.

2025-10-01 10:00

38. An apparatus comprising:
- a hash comprising a bucket with a plurality of locations to store unique keys;
  - a cache to store least recently used unique keys; and
  - a searching module to receive a unique key and to concurrently search the hash and the cache to find a match to the received unique key from the unique keys stored in the hash and the least recently used unique keys stored in the cache.
39. The apparatus of claim 38 wherein the recently used unique keys in the cache are assigned a timestamp.
40. The apparatus of claim 39 wherein the unique key with an oldest timestamp is removed from the cache.
41. The apparatus of claim 38 wherein a first location in the bucket comprises a most recent key.
42. The apparatus of claim 41 wherein the most recent key is a most recently searched key.
43. The apparatus of claim 41 wherein the most recent key is a last added key.



44. A line card comprising:

a first memory to store a plurality of most recently used keys;

a second memory to store a plurality of least recently used keys  
relative to the most recently used keys; and

a processor coupled with the first and second memories, the  
processor to receive a search key and to concurrently search the first and  
second memories for a matching key among the most recently used keys  
and the least recently used keys.

45. The line card of claim 44 wherein the first memory is a random access  
memory.

46. The line card of claim 44 wherein the second memory is a content  
addressable memory.

47. The line card of claim 44 further comprising a third memory coupled with  
the processor, the processor to determine an address of the matching key  
and transmit the address to the third memory.

48. The line card of claim 47 wherein the third memory is configured to store  
a plurality of different entries.

49. The line card of claim 47 wherein the third memory is a synchronous dynamic random access memory.
50. The line card of claim 48 wherein the search key corresponds to one of the plurality of different entries in the third memory.
51. The line card of claim 44 wherein each least recently used key is assigned a timestamp.
52. The line card of claim 51 wherein a recently used key with an oldest timestamp is removed from the second memory.
53. The line card of claim 44 wherein the first memory comprises a plurality of buckets comprising a plurality of locations.
54. The line card of claim 53 wherein the processor is further configured to add a new key to the first memory.
55. The line card of claim 54 wherein the processor is further configured to add a new key to a first location of the plurality of locations of a bucket.

56. The line card of claim 55 wherein the processor is further configured to rotate entries of remaining locations of the plurality of locations of the bucket.
57. The line card of claim 56 wherein the processor is further configured to move an entry from a last location of the plurality of locations of the bucket to the second memory.
58. The line card of claim 53 wherein the processor is further configured to delete the matching key from the first memory.
59. The line card of claim 58 wherein the processor is further configured to rotate entries in remaining locations of the plurality of locations of a bucket.
60. The line card of claim 59 wherein the processor is further configured to the rotate the entries in the remaining locations of the bucket by shifting entries in locations following a deleted matching key location to place an entry stored in a location next to the deleted matching key location into the deleted matching key location.
61. An apparatus comprising:

means for receiving a unique key;  
means for searching a hash for a match to the unique key;  
means for searching a cache for the match to the unique key  
concurrently with searching the hash for the match to the unique key; and  
means for obtaining information regarding the unique key.

62. The apparatus of claim 61 wherein the hash comprises buckets, each  
bucket comprising a plurality of locations.

63. The apparatus of claim 62 wherein a first location in the bucket comprises  
a most recent key.

64. The apparatus of claim 61 wherein the cache comprises least recently used  
keys.

65. A machine-readable medium that provides instructions, which when  
executed by a machine, cause the machine to perform operations  
comprising:

receiving a unique key;  
searching a hash for a match to the unique key;  
searching a cache for the match to the unique key concurrently with  
the searching the hash for the match to the unique key;

and

obtaining information regarding the unique key.

66. The machine-readable medium of claim 65 wherein the hash comprises a plurality of buckets, each bucket comprising a plurality of locations.

67. The machine-readable medium of claim 66 wherein a first location in the bucket comprises a most recent key.

68. The machine-readable medium of claim 65 wherein the cache comprises least recently used keys.

69. A machine-readable medium that provides instructions, which when executed by a machine, cause the machine to perform operations comprising:

rotating entries stored in a plurality of locations of a hash bucket to empty a first location; and

adding a key to the first location of the plurality of locations of the hash bucket.

70. The machine-readable medium of claim 69 wherein the rotating entries comprises moving an entry from a location to a next location.

71. The machine-readable medium of claim 70 further causes the machine to perform operations further comprising moving an entry from a last location to a cache.